

Objective C Programming

Lecture 4
Design patterns. Cocoa intro.

International University of Information Technology
Almaty, 2012

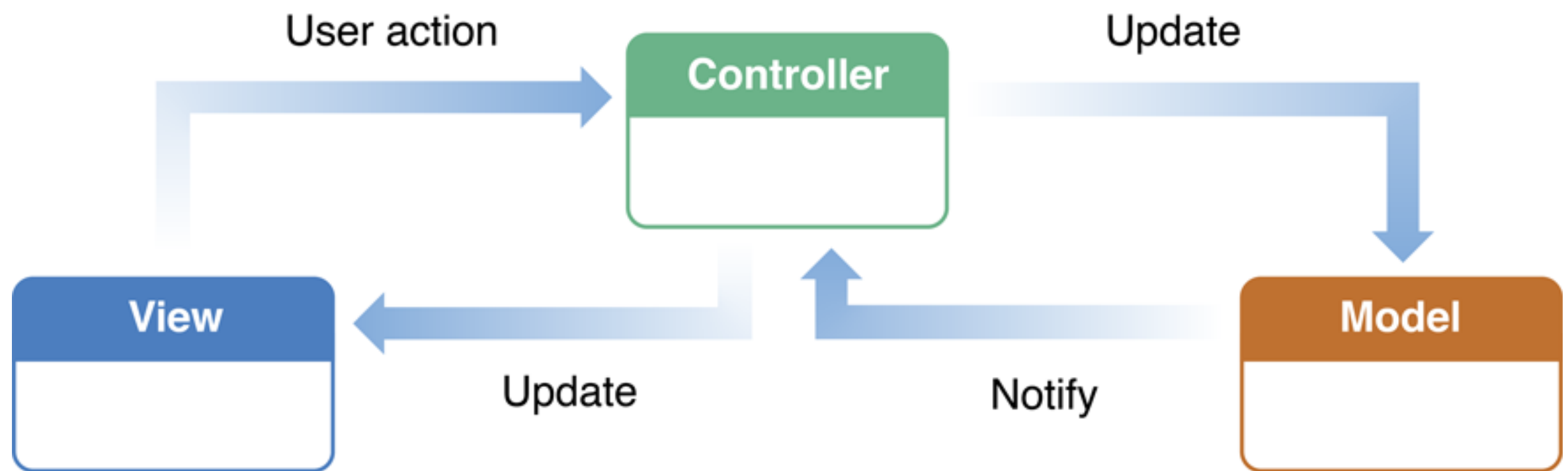
Syllabus fix

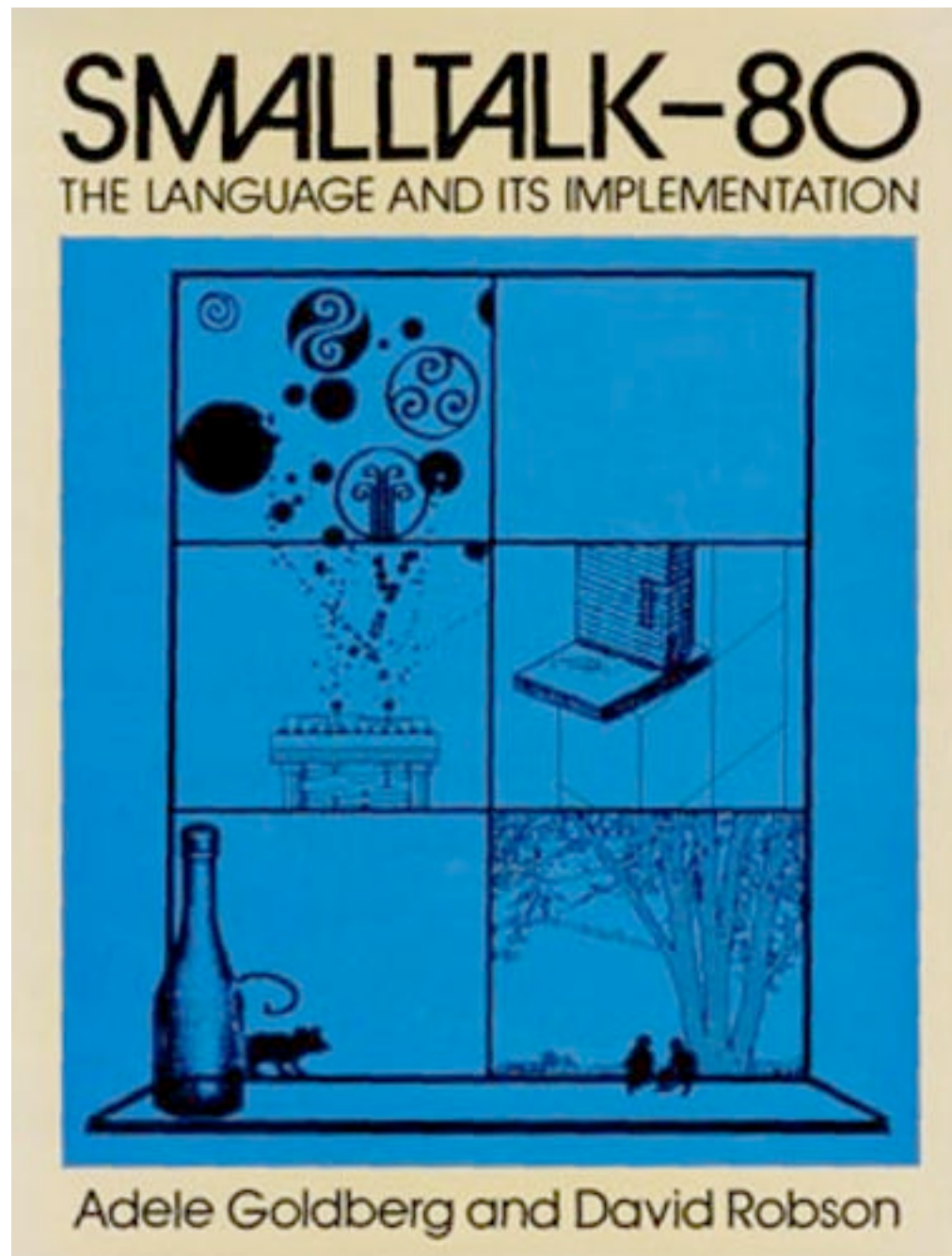
- Core data topic moves to later weeks
- Design patterns and cocoa comes first now
- this is actually good news ;-)

Design pattern

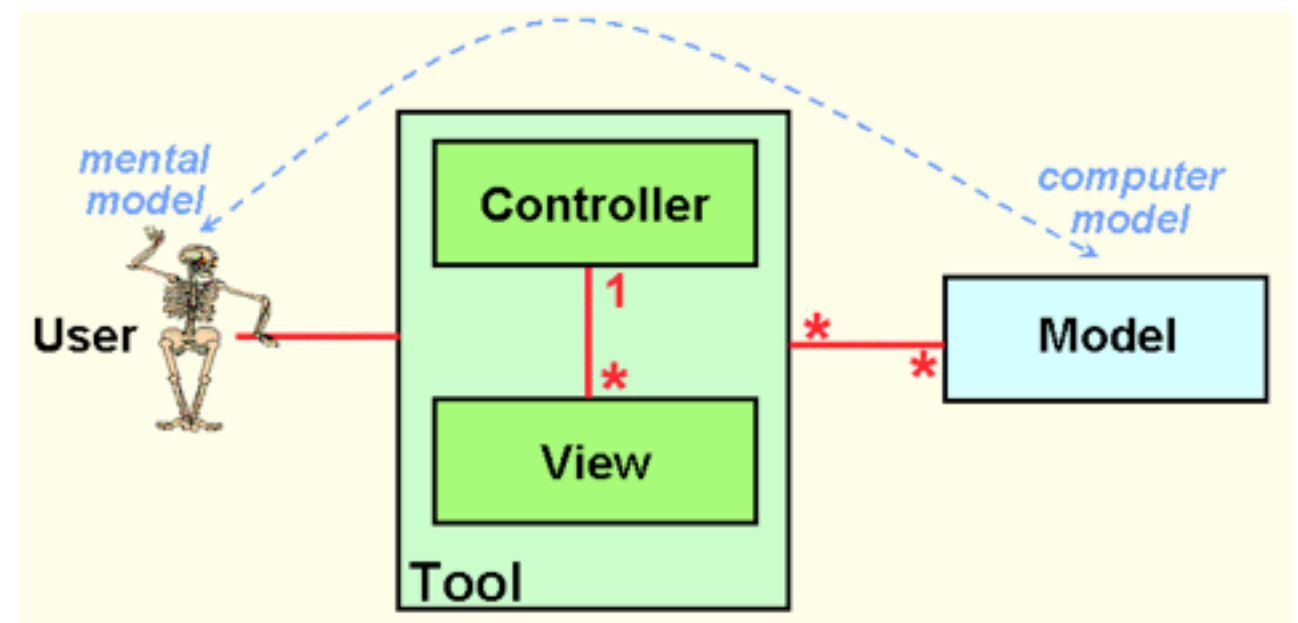
- a general reusable solution to a commonly occurring problem within a given context in software design
- can speed up the development process by providing tested, proven development paradigms

MVC





MVC was invented by one Smalltalk programmer, Trygve Reenskaug.



<http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>

Model

- Data & logic of the app

View

- What user sees and interacts with

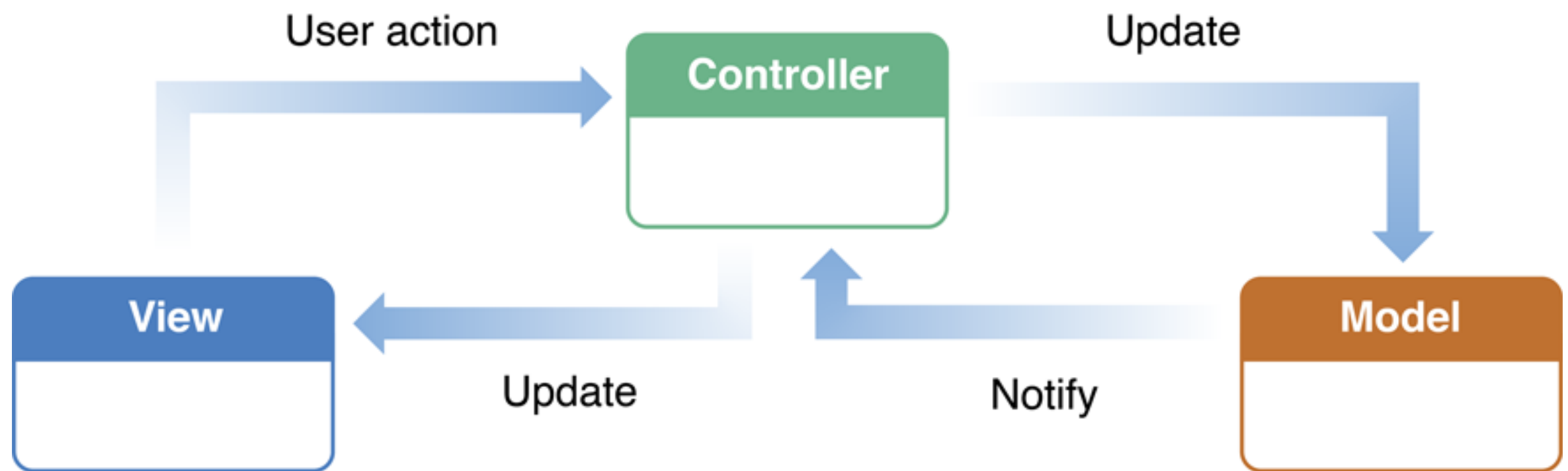
Controller

- Intermediary between one or more of an app's view objects and one or more of its model objects

Quick test

Here's one quick way to test if your application has properly segregated itself between the Model, View, and Controller roles: **is your app skinnable?**

MVC

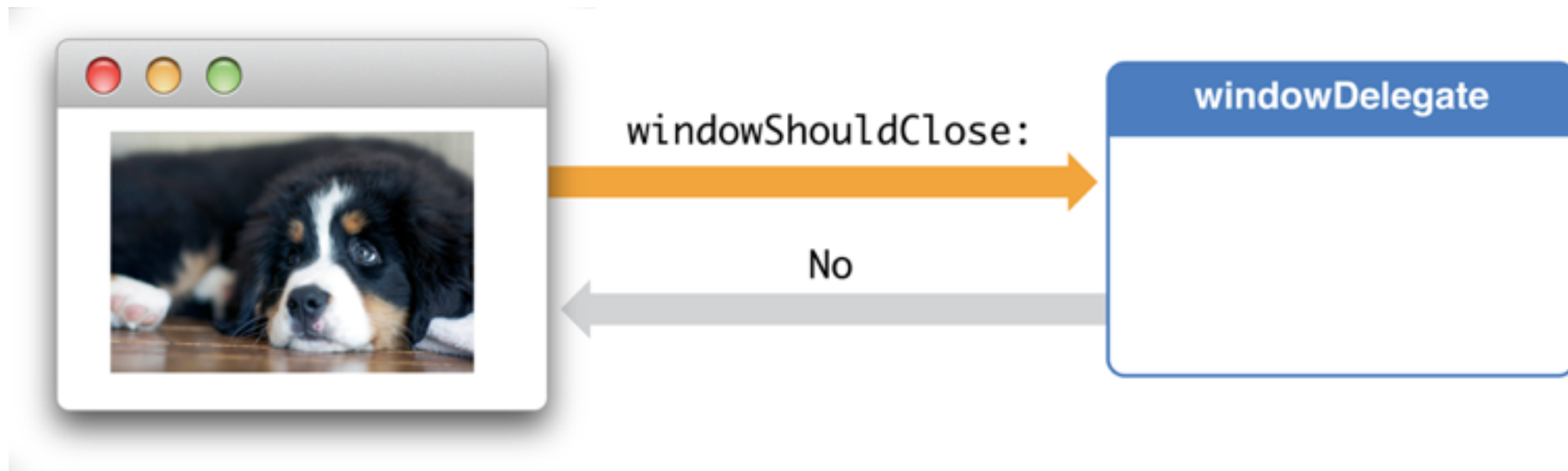


Solving problems

This is a brief overview.

Delegation

- an object called the **delegate** acts on behalf of, and at the request of, another object



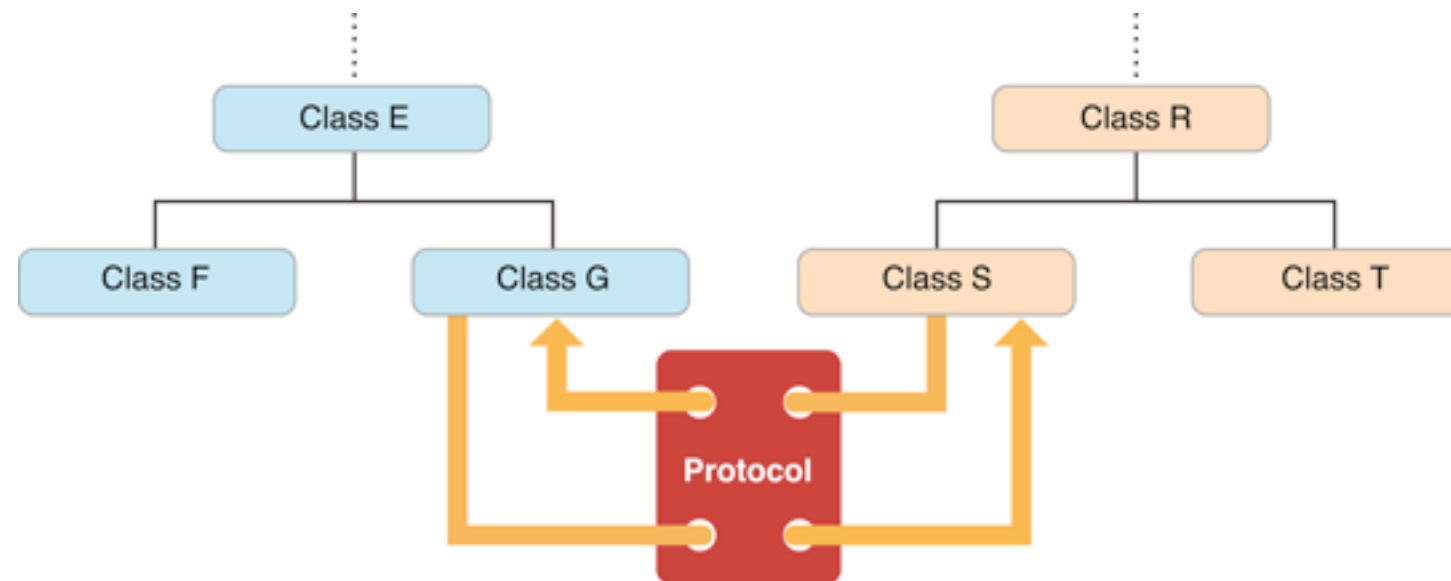
Delegation is a means for injecting application-specific behavior into the workings of a framework class, without having to subclass that class.

Delegating class should

- Define a property (“delegate”) to hold the reference to the delegate
- Define a protocol that the class of the delegate should adopt

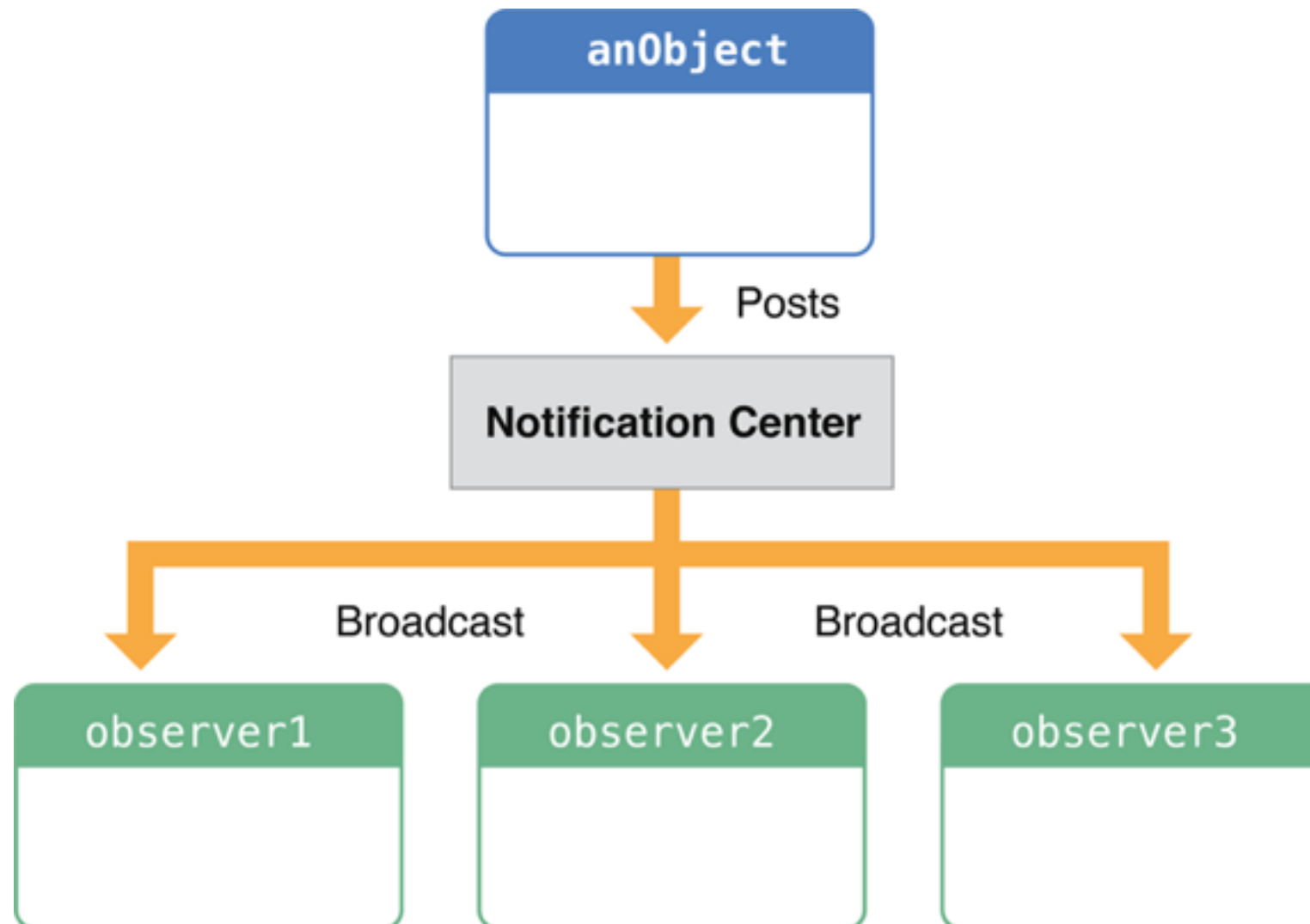
Protocols, right?

Enabling communication between objects that are unrelated by inheritance



Notification Center

A subsystem of the Foundation framework that broadcasts a message – a notification – to all objects in an app that are registered observers of an event



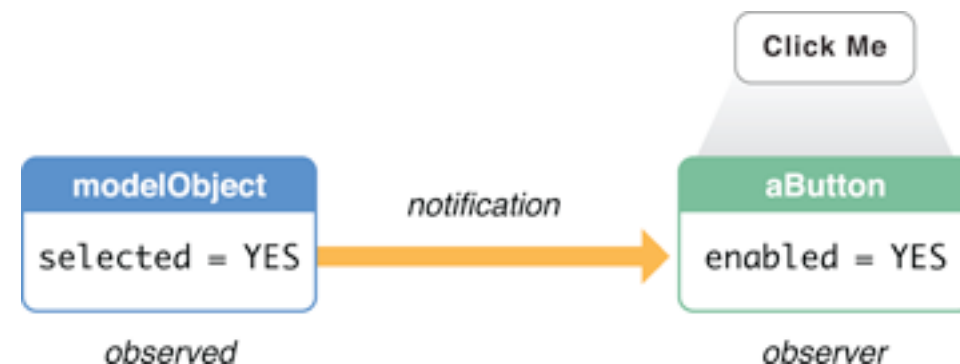
Similarities


Notification-center notifications are similar to delegation messages; both are sent to arbitrary objects when certain events occur.

However, methods that handle notifications, unlike delegation methods, cannot return a value.

Key-Value Observing

- KVO allows an object to observe a property of another object
- The observing object is notified when that property's value changes. It learns about the old value as well as the new one





This is how we
build apps

I'm scared...

COCOA

How we proceed

- There are few about Foundation Framework things we haven't discussed
- We will cover them as we discover Cocoa
- And learn Core Data when we become confident with Cocoa and Foundation

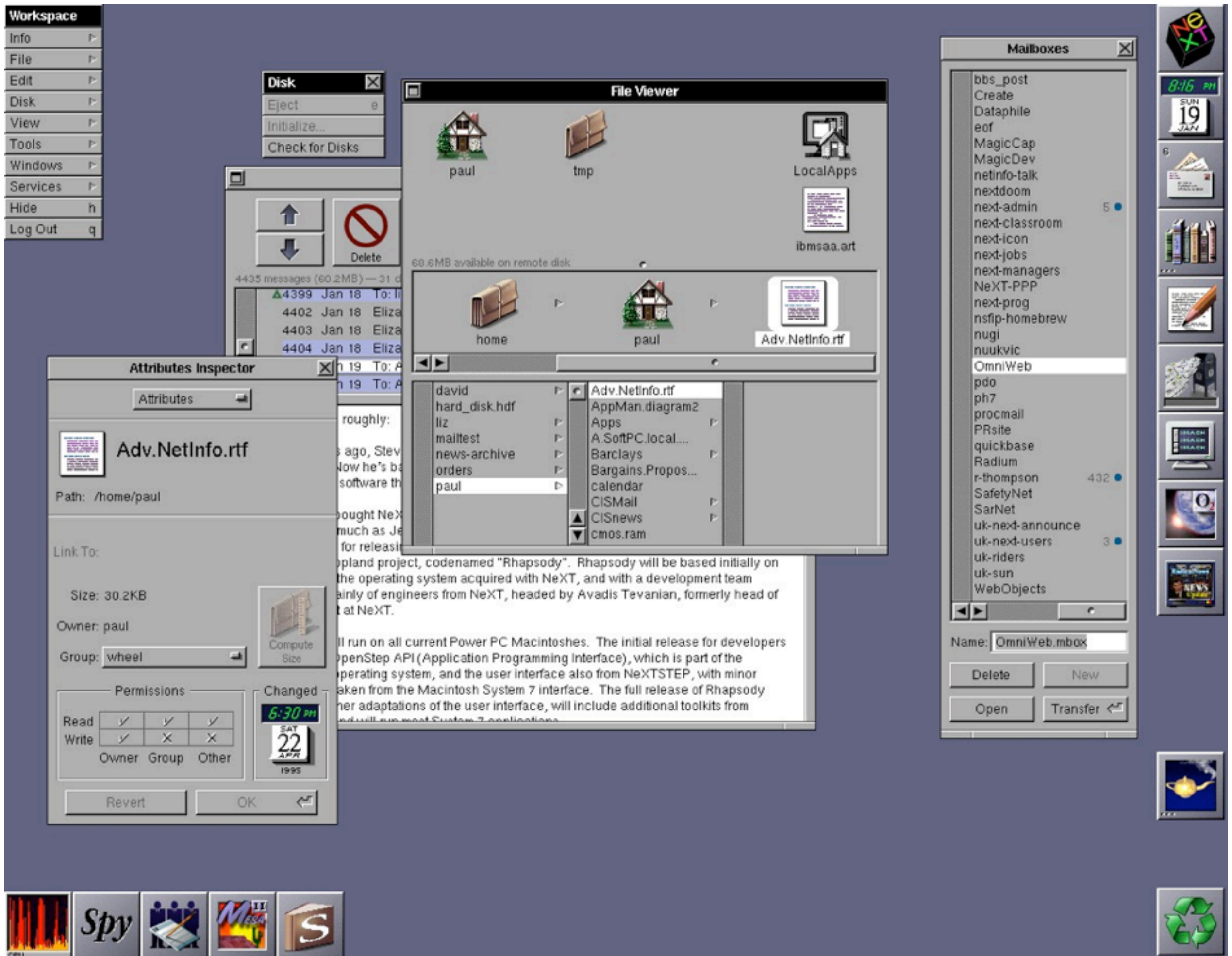
Little bit of history



PHOTO: DIANE COOK AND LEN JEL







NeXTSTEP

|

OpenStep

|

Cocoa

Let's try something

Debug

Projects

- Assignment I is due next week (Oct. 3)
- Assignment II will be the start of your own project, which you will build during assignments II, III and IV.